REMARKS

Claims 1-5 and 21-35 are pending in this Application, of which Claims 1 and 21 are the independent claims. All claims stand rejected.

Rejection of Claims 1-5 and 21-35 under 35 U.S.C. § 102

Claims 1-5 and 21-35 stand rejected under 35 U.S.C. § 102(b) as being anticipated by U.S. Patent No. 6,151,602 to Hejlsberg *et al.* (hereinafter, "Hejlsberg"). Claim 1 is amended in the claim listing above to recite that the data parser determines field boundaries in the non-field delineated database records and processes field to select one or more fields to be assembled into output tuples. Claim 21 is similarly amended. Support for these amendments may be found at least in paragraphs [0017] and [0020] of Applicants' published application (No. 2004/0139214).

Applicants respectfully disagree with the aforementioned rejections in view of the Claims as currently amended, and reconsideration is respectfully requested.

Without limitation to the claims, embodiments of the present invention relate to a data engine, including a data parser that can be programmed to recognize the record and field structures of non-field delineated data. With this structural information regarding the non-field delineated data, the data parser determines field boundaries in the non-field delineated data and then parses the non-field delineated data into field delineated data. The data parser then processes fields to select one or more fields to be assembled into output tuples. The data engine compares fields to precisely determine which records are worth selecting to be transferred to memory for further processing. An output tuple is then formed comprised of the fields of the source record that are to be selected for further processing. Thus, the data engine performs certain preliminary processing in order to reduce the computational load required for the further processing.

The Office Action, in the "Response to Arguments" on page 7, states that neither Applicants nor the specification provide a definition for "non-field delineated" data and, therefore, the Office interprets "non-field delineated data" as "data without field delineated." According to the Microsoft Computer Dictionary (Microsoft Press; 5 ed.; June 1, 2002), a field is "A location in a record in which a particular type of data is stored. In relational database management systems, fields are called columns" (see Exhibit I, enclosed). Further, as

understood in the art, delineation requires the use of a delimiter, defined as "A special character that sets off, or separates, individual items in a...set of data." Examples of field delineated data are provided in paragraph [0076] of Applicants' specification, and include tables, indices and views. Therefore, one of ordinary skill in the art would appreciate that, unlike the example field-delineated data, non-field delineated data is not arranged in columns and does not include delimiters between items of data.

Further, the Office Action states on p. 7, "if a reference does not teach 'field delineated data', then the reference anticipates the claimed 'non-field delineated data." However, this conclusion is inconsistent with 35 U.S.C. § 102(b) which requires that a reference teach each element of a claim, rather than require that a reference not teach that which is not claimed as suggested in the Office Action.

Hejlsberg relates to a data packet that may be transmitted between a client and a server in a three-tier data processing system. An example three-tier data processing system 300 shown in FIG. 3 comprises a client 310, a server 350 and a middle tier, which includes a provider 320 and a resolver 325. The data packet replicates a data set from the server to the client. With reference to FIG. 3 (in conjunction with the flow diagram 500 of FIG. 5), as described at col. 6, lines 36-58, to obtain data, the client 310 sends a request (501) for data from a data source to the middle tier (provider) 320 which, in turn (502), retrieves the data from the server. The request is honored by the provider 320 generating and returning a data "snapshot" of the result data set (503-507), which is stored locally 315 at the client in the form of a self-describing data packet, as described in col. 3, lines 7-16 and 27-45, that represents a results set received by a client from a data source.

However, Hejlsberg neither determines field boundaries in non-field delineated database records nor parses the non-field delineated database records into field delineated data, as now recited in amended Claims 1 and 21. Moreover, Hejlsberg cannot perform such operations because, contrary to the assertions in the Office Action, the data in Hejlsberg is field delineated at all times. As described below, the data in Hejlsberg is field delineated (1) at the server; (2) in its packetized form for transmission; and (3) as unpacked at the client.

(1) At the server, for example, the provider 320 retrieves field-delineated data from a database server. That field-delineated data is then transformed by the provider 320 into a self-describing data packet, as described below.

(2) During transmission, as illustrated in FIG. 4 and described at col. 8, lines 14-23 and 36-48, self-describing packet 400 contains "a variable number of rows 430 containing the actual row data." Therefore, as taught at col. 20, lines 48-52 and 64 through col. 21, line 1 of Hejlsberg, "the provider can sequentially read column descriptor information from the data source...and then stream out corresponding metadata...Processing of actual data also occurs sequentially, as the information is being written out to the stream. In particular, the system loops through all data records of the result set and writes or streams out the corresponding field values, at step 505. Here, only actual data up [sic] is written out." Thus, the rows of data in the self-describing data packet include the actual row data of the data set, except that the column description information has been stripped off into metadata for reconstitution at the client. As described at col. 3, lines 36-40, this allows compact data transmission. A row of data is, by definition, field-delineated in columns.

(3) The row data is still field-delineated at the client where it is unpacked from the self-describing data packet into a reconstituted record to include the column descriptor information. Referring to FIG 3 (in conjunction with the flow diagram 600 of FIG. 6), as described at col. 7, lines 39-54 and col. 21, lines 16-26, the structure of the data in a self-describing data packet is described using metadata. Upon receiving the self-describing data packet from the provider 320, the client 310 unpacks (601) the data. Through use of the metadata (602, 603), the client has full knowledge of the data set. The data then is stored locally (604) and processed as if it originally were local data (605).

Therefore, Hejlsberg fails to teach non-field delineated data and, consequently, cannot determine field boundaries in non-field delineated data nor parse non-field delineated data into field delineated data, as now recited in amended Claims 1 and 21.

Further, even if Hejlsberg did teach non-field delineated data, Hejlsberg fails to teach processing fields to select one or more fields to be assembled into output tuples, as required by the independent claims as amended, and an output tuple generator configured to assemble filtered field delineated data into an output tuple. The Office Action relies on col. 21, lines 25-35

and 47 through col. 22, line 25 of Hejlsberg for such a teaching. However, that portion of Hejlsberg teaches that the reconstituted database records exist at the client as if it were a local table and that the reconstituted database records are exactly the same as the database records as they exist at the server end of the system. Therefore, Hejlsberg fails to teach any selection of fields for further processing and the claimed output tuples into which those fields are assembled. Although filtering of some form may be performed in Hejlsberg, such data manipulations are performed by the client, thus requiring that the reconstituted database records have been transmitted already to the client. Therefore, these manipulations in Hejlsberg occur after the data packet has been transferred, not prior to the assembly of an output tuple.

In order to anticipate a claim, a reference must teach each and every element of the claim. For the reasons presented above, Hejlsberg fails to teach the elements of independent Claim 1. Therefore, the rejection of independent Claim 1 is overcome and reconsideration is respectfully requested. Similarly, with regard to the rejection of independent Claim 21, Hejlsberg does not teach the claimed method that substantially corresponds to the data engine of Claim 1. For these reasons, the rejections of independent Claims 1 and 21 are overcome and reconsideration is respectfully requested.

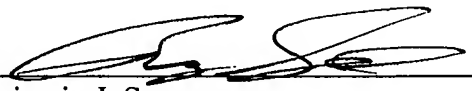
Claims 2-5 and 22-35 depend from Claims 1 and 21, and therefore inherit the limitations of the respective base claims. As a result, the rejections of Claim 2-5 and 22-35 are overcome and reconsideration is respectfully requested.

CONCLUSION

In view of the above amendments and remarks, it is believed that all claims are in condition for allowance, and it is respectfully requested that the application be passed to issue. If the Examiner feels that a telephone conference would expedite prosecution of this case, the Examiner is invited to call the undersigned.

Respectfully submitted,

HAMILTON, BROOK, SMITH & REYNOLDS, P.C.

By 

Benjamin J. Sparrow
Registration No. 62,259
Telephone: (978) 341-0036
Facsimile: (978) 341-0136

Concord, MA 01742-9133
Date: 10/2/09